UNIT-1 Knowledge-Representation and Reasoning

Logical Agents: Knowledge based agents, the Wumpus world, logic. Patterns in propositional Logic, inference in Fist-Order Logic propositional vs fist order inference unification and lifting

---

Logical Agents in AI & ML

Logical agents are **intelligent systems** that use **logic** to make decisions. They **store knowledge**, **reason about it**, and **take actions** based on logical rules.

---

1. What is a Logical Agent?

A **logical agent** is a type of AI that:
✅ Uses **knowledge** to understand the world.
✅ Uses **logical reasoning** to make decisions.
✅ Can **infer** new information from known facts.
✅ Takes **actions** based on logical conclusions.

🔷 **Example:**
Imagine a **robot vacuum cleaner**:

- It **knows** that "Dirt is bad".
- It **sees dirt** in a room.
- It **decides** to clean the dirt.
- It **acts** by moving towards the dirt and vacuuming.

Logical agents **think logically**, just like humans solving puzzles! 🧠

---

2. Knowledge-Based Agents

A **knowledge-based agent (KBA)** is a type of logical agent that:

- **Has a knowledge base (KB)** → Stores facts about the world.
- **Uses inference rules** → Applies logic to make decisions.
- **Updates its knowledge** → Learns new information over time.

Components of a Knowledge-Based Agent:

1. **Knowledge Base (KB):** Stores facts and rules.
2. **Inference Engine:** Uses logic to reason and make decisions.
3. **Perception:** Observes the environment.
4. **Action Execution:** Takes actions based on logical conclusions.

🔷 **Example:**
A medical diagnostic system (like **IBM Watson**) works as a **knowledge-based agent**:

- **KB:** Stores medical facts (e.g., "Fever + Cough → Maybe Flu").
- **Inference:** If a patient has fever & cough, **concludes flu**.
- **Action:** Suggests medicine or further tests.

---

3. The Wumpus World (A Logical Agent Game)

The **Wumpus World** is a classic **AI problem** used to test logical agents. It is a **grid-based game** where an agent must find **gold** while avoiding **dangers**.

The Environment:

🔷 The world is a **4x4 grid** with:
✅ **Gold** → The goal (agent must find it).
❌ **The Wumpus** → A dangerous monster (agent must avoid it).
⚠️ **Pits** → Deadly holes (agent must avoid them).
🌬️ **Breeze** → Found near pits (helps detect danger).
💨 **Stench** → Found near the Wumpus (helps locate it).

Agent's Knowledge & Reasoning:

1. **Starts in the bottom-left corner (safe square).**
2. **Moves through the grid, sensing breeze & stench.**
3. **Uses logic to decide where to go safely.**
4. **Grabs the gold & exits safely.**

---

4. How Logical Agents Solve the Wumpus World

🔷 The agent follows logical **rules** to make decisions:

✅ Example Rule 1: If there is a breeze, a pit might be nearby.
✅ Example Rule 2: If there is a stench, the Wumpus is nearby.
✅ Example Rule 3: If a square has no danger signs, it is safe to move.

🧠 **Logical Deduction Example:**

- The agent moves into a square and senses **breeze**.
- It **infers** that a pit must be nearby.
- It **marks the dangerous area** and avoids moving there.
- It **continues exploring** and eventually finds the **gold safely**.

🔷 **Real-Life Analogy:**
The Wumpus World is like **a blindfolded treasure hunt**:

- You **hear sounds** (stench, breeze) to guess dangers.
- You **use reasoning** to avoid danger.
- You **find the gold** using logic!

---

5. Applications of Logical Agents

Logical agents are used in real-world AI applications:

✅ **Medical Diagnosis:** AI doctors use logical reasoning to diagnose diseases.
✅ **Game AI:** Chess and video game characters use logic to plan moves.
✅ **Autonomous Vehicles:** Self-driving cars use logical rules to navigate safely.
✅ **Virtual Assistants:** Siri, Alexa, and Google Assistant answer questions logically.

Patterns in Propositional Logic in AI & ML

1. Introduction to Propositional Logic in AI & ML

Propositional Logic (PL) is a fundamental part of **Artificial Intelligence (AI) and Machine Learning (ML)**. It is used to represent **knowledge** and make **logical inferences** about facts.

What is Propositional Logic?

- A **formal system** that uses **propositions (statements)** to express **truth values** (True or False).
- Used in **Knowledge-Based Systems, Expert Systems, and Logical Agents**.
- Forms the foundation of **automated reasoning** in AI.

🔷 **Example:**

- **Statement (Proposition):** "If it is raining, the ground is wet."
- **Logic Representation:**
  - Let **P = "It is raining"**
  - Let **Q = "The ground is wet"**
  - **Rule:** P⇒QP \Rightarrow QP⇒Q (If P is true, then Q must also be true).

---

2. Basic Elements of Propositional Logic

2.1 Propositions (Statements)

- **A Proposition** is a **statement that is either True or False**.
- **Example Propositions:**
  - **P:** "It is raining." ✅ (Can be True or False)
  - **Q:** "The ground is wet." ✅ (Can be True or False)
  - "What time is it?" ❌ (Not a proposition – it is a question, not a statement).

2.2 Logical Operators (Connectives)

Propositional logic uses **logical operators** to combine statements.

Symbol	Name	Meaning
¬P	**Negation (NOT)**	"It is NOT raining."
P ∧ Q	**Conjunction (AND)**	"It is raining AND the ground is wet."
P ∨ Q	**Disjunction (OR)**	"It is raining OR the ground is wet."
P ⇒ Q	**Implication (IF-THEN)**	"If it is raining, THEN the ground is wet."
P ⇔ Q	**Biconditional (IF AND ONLY IF)**	"The ground is wet IF AND ONLY IF it is raining."

---

3. Patterns in Propositional Logic

Patterns are **common logical structures** that help in **knowledge representation and inference**.

3.1 Modus Ponens (Pattern 1: Direct Reasoning)

🔷 **Definition:**
If **P ⇒ Q** is true, and **P** is true, then **Q** must also be true.

🔷 **Example:**

1. **If it is raining, the ground is wet.** (P ⇒ Q)
2. **It is raining.** (P is True)
3. **Conclusion: The ground is wet.** (Q is True)

🔷 **Logic Representation:**

(P⇒Q),P⊢Q(P ⇒ Q), P ⊢ Q(P⇒Q),P⊢Q

🔷 **Real-Life Example:**

- **Rule:** "If a person studies hard, they will pass the exam."

- **Fact:** "John studied hard."
- **Conclusion:** "John will pass the exam."

3.2 Modus Tollens (Pattern 2: Indirect Reasoning)

◆ **Definition:**
If **P ⇒ Q** is true, and **Q is false**, then **P must also be false**.

◆ **Example:**

1. **If it is raining, the ground is wet.** (P ⇒ Q)
2. **The ground is NOT wet.** (¬Q)
3. **Conclusion: It is NOT raining.** (¬P)

◆ **Logic Representation:**

(P⇒Q),¬Q⊢¬P(P ⇒ Q), ¬Q ⊢ ¬P(P⇒Q),¬Q⊢¬P

◆ **Real-Life Example:**

- **Rule:** "If the car has fuel, it will start."
- **Fact:** "The car did NOT start."
- **Conclusion:** "The car does NOT have fuel."

3.3 Hypothetical Syllogism (Pattern 3: Chain Reasoning)

◆ **Definition:**
If **P ⇒ Q** and **Q ⇒ R**, then **P ⇒ R**.

◆ **Example:**

1. **If it is raining, the ground is wet.** (P ⇒ Q)
2. **If the ground is wet, people carry umbrellas.** (Q ⇒ R)
3. **Conclusion: If it is raining, people carry umbrellas.** (P ⇒ R)

◆ **Logic Representation:**

(P⇒Q),(Q⇒R)⊢(P⇒R)(P ⇒ Q), (Q ⇒ R) ⊢ (P ⇒ R)(P⇒Q),(Q⇒R)⊢(P⇒R)

◆ **Real-Life Example:**

- **Rule 1:** "If a student studies, they pass the exam."
- **Rule 2:** "If they pass the exam, they get a certificate."
- **Conclusion:** "If a student studies, they get a certificate."

3.4 Disjunctive Syllogism (Pattern 4: Eliminating Options)

🔷 **Definition:**
If **P ∨ Q** is true, and **P is false**, then **Q must be true**.

🔷 **Example:**

1. **Either the lamp is broken OR the bulb is fused.** (P ∨ Q)
2. **The lamp is NOT broken.** (¬P)
3. **Conclusion: The bulb is fused.** (Q)

🔷 **Logic Representation:**

(P∨Q),¬P⊢Q(P ∨ Q), ¬P ⊢ Q(P∨Q),¬P⊢Q

🔷 **Real-Life Example:**

- **Statement:** "Either John is at home OR he is at work."
- **Fact:** "John is NOT at home."
- **Conclusion:** "John is at work."

3.5 Resolution (Pattern 5: Combining Knowledge)

🔷 **Definition:**
If **P ∨ Q** and **¬Q ∨ R** are true, then we can infer **P ∨ R**.

🔷 **Example:**

1. **Either it is raining OR it is sunny.** (P ∨ Q)
2. **Either it is NOT sunny OR people go outside.** (¬Q ∨ R)
3. **Conclusion: Either it is raining OR people go outside.** (P ∨ R)

🔷 **Logic Representation:**

(P∨Q),(¬Q∨R)⊢(P∨R)(P ∨ Q), (¬Q ∨ R) ⊢ (P ∨ R)(P∨Q),(¬Q∨R)⊢(P∨R)

🔷 **Real-Life Example:**

- **Statement 1:** "Either the battery is dead OR the phone is on silent."
- **Statement 2:** "Either the phone is NOT on silent OR you will hear a ringtone."
- **Conclusion:** "Either the battery is dead OR you will hear a ringtone."

---

4. Applications of Propositional Logic Patterns in AI & ML

🔷 **Expert Systems (Medical Diagnosis, Legal AI)**

- AI doctors use **Modus Ponens** to diagnose diseases.
- Example: "If a patient has a fever, they might have the flu."

🔷 **Game AI (Chess, Wumpus World)**

- Uses **Disjunctive Syllogism** to eliminate bad moves.

🔷 **Robotics & Autonomous Vehicles**

- Uses **Hypothetical Syllogism** for navigation.
- Example: "If there is an obstacle ahead, the car should turn."

🔷 **Natural Language Processing (NLP)**

- Uses **Resolution** to infer meaning from sentences.

Inference in First-Order Logic (FOL) – Step by Step Explanation

1. Introduction to Inference in First-Order Logic (FOL)

Inference in **First-Order Logic (FOL)** is the process of **deriving new facts** from **known facts and rules**. It allows AI systems to **reason logically** about the world.

What is First-Order Logic (FOL)?

First-Order Logic (also called **Predicate Logic**) extends **Propositional Logic** by adding:
✅ **Quantifiers** – Express statements about "all" or "some" objects.
✅ **Predicates** – Describe relationships between objects.
✅ **Variables** – Represent unknown entities.

🔷 **Example:**

- **Statement in English:** "All humans are mortal."
- **FOL Representation:** ∀x Human(x)⇒Mortal(x)\forall x \, \text{Human}(x) \Rightarrow \text{Mortal}(x)∀xHuman(x)⇒Mortal(x)
    - **∀ x** (For all x): Applies to all objects.
    - **Human(x):** x is a human.
    - **Mortal(x):** x is mortal.

🔷 **Inference Goal:**
Given **"Socrates is a human"**, can we infer that **"Socrates is mortal"**?

---

2. Propositional Logic vs. First-Order Logic (Differences in Inference)

Feature	**Propositional Logic (PL)**	**First-Order Logic (FOL)**
Variables	❌ No variables	✅ Uses variables (x, y, etc.)
Quantifiers	❌ No quantifiers	✅ Uses **∀ (for all)** and **∃ (there exists)**
Expressiveness	✅ Simple	🔥 More powerful
Example Rule	"If it rains, the ground is wet"	"If a person is a student, they have a teacher"
Limitation	Cannot generalize	Can represent relationships and general knowledge

🔷 **Example Comparison:**

□Propositional Logic (PL):

- **Rule:** "If it rains, the ground is wet."
  - **P⇒QP \Rightarrow QP⇒Q** (where P = "It rains", Q = "The ground is wet")
- **Fact:** "It rains."
- **Inference:** "The ground is wet."

2□First-Order Logic (FOL):

- **Rule:** "If a student is enrolled in a class, they have a teacher."
  - **∀x Student(x)⇒HasTeacher(x)\forall x \, \text{Student}(x) \Rightarrow \text{HasTeacher}(x)∀xStudent(x)⇒HasTeacher(x)**
- **Fact:** "Alice is a student." (**Student(Alice)**)
- **Inference:** "Alice has a teacher." (**HasTeacher(Alice)**)

Why FOL is More Powerful?

FOL can **express knowledge more generally** using variables and quantifiers, whereas **PL requires separate rules for each individual case**.

---

3. Inference in First-Order Logic (FOL) – Step by Step

Inference in FOL follows these methods:
✅ **Forward Chaining** – Start with known facts and apply rules to infer new facts.
✅ **Backward Chaining** – Start with a goal and work backward to prove it.
✅ **Resolution** – Use logical contradiction to derive conclusions.
✅ **Unification & Lifting** – Match variables with constants or other variables.

Step 1: Forward Chaining (Data-Driven Inference)

- Starts with **known facts** and applies rules to **derive new conclusions**.
- Used in **expert systems** and **rule-based AI**.

🔷 **Example:**

1. **Rule:** "If x is a bird, then x can fly." ∀x Bird(x)⇒CanFly(x)\forall x \, \text{Bird}(x) \Rightarrow \text{CanFly}(x)∀xBird(x)⇒CanFly(x)
2. **Fact:** "Tweety is a bird." (**Bird(Tweety)**)
3. **Inference:** "Tweety can fly." (**CanFly(Tweety)**)

✅ **AI Application:** Used in **diagnostic expert systems** (e.g., medical AI).

Step 2: Backward Chaining (Goal-Driven Inference)

- Starts with a **goal** and tries to prove it using known facts.
- Used in **AI Planning & Logical Programming** (e.g., Prolog).

🔷 **Example:**

1. **Query:** "Can Tweety fly?" (**CanFly(Tweety)?**)
2. **Rule:** "If x is a bird, then x can fly."
3. **Fact:** "Tweety is a bird."
4. **Conclusion:** "Yes, Tweety can fly."

✅ **AI Application:** Used in **chatbots and automated reasoning systems**.

Step 3: Resolution (Proving by Contradiction)

- Converts **FOL statements into clauses** and **derives contradictions** to prove the conclusion.
- Used in **automated theorem proving**.

🔷 **Example:**

1. **Given Statements:**
   - ∀x Student(x)⇒Studies(x)\forall x \, \text{Student}(x) \Rightarrow \text{Studies}(x)∀xStudent(x)⇒Studies(x) (All students study).
   - **NotFact:** ¬Studies(Alice)\neg \text{Studies}(Alice)¬Studies(Alice) (Alice does not study).
2. **Contradiction Found:** Alice **is a student but does not study**.

3. **Conclusion: Reject "Alice does not study"** → Alice must study.

✔ **AI Application:** Used in **mathematical proofs and AI logic solvers**.

Step 4: Unification & Lifting (Pattern Matching in Inference)

Unification is the process of **matching logical variables** to **constants or other variables** to make inference possible.

4.1 Unification (Finding Matching Substitutions)

🔷 **Example:**

- **Rule:** "All birds can fly." ∀x Bird(x)⇒CanFly(x)\forall x \, \text{Bird}(x) \Rightarrow \text{CanFly}(x)∀xBird(x)⇒CanFly(x)
- **Fact:** "Tweety is a bird." (**Bird(Tweety)**)
- **Unification:** x=Tweetyx = \text{Tweety}x=Tweety
- **Inference:** "Tweety can fly." (**CanFly(Tweety)**)

✔ **Application:** Used in **AI search engines and expert systems**.

4.2 Lifting (Applying Unification to Complex Sentences)

🔷 **Example:**

- **Rule:** "If someone is a parent, they have a child." ∀x Parent(x)⇒∃y Child(x,y)\forall x \, \text{Parent}(x) \Rightarrow \exists y \, \text{Child}(x, y)∀xParent(x)⇒∃yChild(x,y)
- **Fact:** "John is a parent." (**Parent(John)**)
- **Lifting Unification:** x=Johnx = Johnx=John and introduces **y (child's name unknown)**
- **Inference:** "John has a child." (**Child(John, y)**)

✔ **Application:** Used in **Natural Language Processing (NLP) and AI knowledge graphs**.

---

4. Summary of Key Concepts

Concept	Definition	Example
Forward Chaining	Start from facts and apply rules to infer new facts.	"Bird(Tweety) → CanFly(Tweety)"
Backward Chaining	Start with a goal and work backward to prove it.	"Can Tweety fly?" (Check Bird(Tweety))

Concept	Definition	Example
Resolution	Uses logical contradiction to prove a conclusion.	"If Alice is a student, she studies. Alice does not study → Contradiction!"
Unification	Matches variables to constants.	"x = Tweety" in Bird(x) → CanFly(x)
Lifting	Generalizes unification for complex cases.	"John is a parent → John has a child."

---

5. Conclusion

🚀 **First-Order Logic (FOL) allows AI to make complex inferences using variables, predicates, and quantifiers.**
✅ **Forward & Backward Chaining** → Used in **Expert Systems & AI Planning**.
✅ **Resolution** → Used in **Automated Theorem Proving**.
✅ **Unification & Lifting** → Used in **AI Search Engines & NLP**.

Would you like a **Python implementation** demonstrating **FOL inference**? 😊